Occlusion Management Techniques for the Visualization of Transportation Networks in Virtual 3D City Models

Matthias Trapp
Hasso Plattner Institute,
Faculty of Digital Engineering,
University of Potsdam, Germany
matthias.trapp@hpi.de

Fabian Dumke
Hasso Plattner Institute,
Faculty of Digital Engineering,
University of Potsdam, Germany
fabian.dumke@student.hpi.de

Jürgen Döllner
Hasso Plattner Institute,
Faculty of Digital Engineering,
University of Potsdam, Germany
juergen.doellner@hpi.de

ABSTRACT

Transportation networks are important components of digital maps. The display of streets and railroad lines are a key for orientation and navigation using traditional and digital maps. While 2D digital maps enable effective communication, their visualization within 3D digital maps, based on virtual 3D city models, is often subject to occlusion that hinders the effective perception. There are a number of approaches to partially resolve occlusion effects within 3D geovirtual environments such as ghosted and cut-away views as well as interactive multi-perspective views are often applied. This paper proposes occlusion hints to emphasize occluded parts of transportation networks. In contrast to existing occlusion management techniques, these present only hints towards occluded parts to the viewer.

CCS CONCEPTS

• **Human-centered computing** → *Visualization techniques*; • **Computing methodologies** → *Non-photorealistic rendering*; *Graphics processors*.

KEYWORDS

Occlusion Management Techniques, Transportation Networks, Virtual 3D City Models

ACM Reference Format:
Matthias Trapp, Fabian Dumke, and Jürgen Döllner. 2019. Occlusion Management Techniques for the Visualization of Transportation Networks in Virtual 3D City Models. In *Proceedings of The 12th International Symposium on Visual Information Communication and Interaction (VINCI'19).* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Digital 3D maps have become fundamental tools in economy and society that enable the visualization of and interaction with customized, multi-layered geographic information [25]. They can be used in a variety of applications such as navigation systems, traffic monitoring, and as scenery for data visualization. In order to interact with digital maps, support for effective orientation, navigation, and exploration are essential requirements. With respect to this, transportation networks constitute an important asset for orientation and navigation.

However, besides perspective foreshortening, the main drawback of using 3D perspective projections is occlusion. In the context of 3D digital maps, this drawback becomes more immanent, since features of transportation networks are quite likely to be occluded. This is especially true for virtual cameras in near ground perspectives. However, in computer graphics and visualization, there are a number of general approaches to manage or counterbalance occlusions in 3D [12] (cf. Section 2), but only few are specific to transportation networks and virtual 3D city models in particular.

This paper describes and discusses approaches for Occlusion Management Techniques (OMTs) suitable for transportation networks in the context of virtual 3D city models. The presented techniques can be applied as assistive interaction techniques for digital 3D Maps, e.g., for highlighting of routes. It briefly describes their implementation and discusses their application. In particular, it presents two novel OMTs: *Fence Occlusion Hints (FOHs)* and *Outline Occlusion Hints (OOHs)*. These techniques do not completely resolve occlusions but give the viewer a *hint* which parts of the transportation network are occluded.

The remainder of this paper is structured as follows. Section 2 reviews related and previous work for approaches that are capable of reducing occlusion with 3D virtual city models. Section 3 presents the application and the respective rendering approaches as well as discusses existing and novel occlusion management approaches for transportation networks. Section 4 briefly describes aspects of a prototypical implementation. Finally, Section 5 concludes this paper and present ideas for future research.

2 RELATED WORK

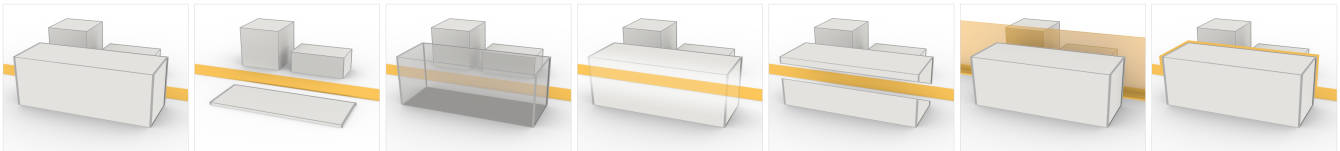
Besides different GPU-based rendering and stylization techniques for transportation networks [35], there is a vast body of work with respect to general occlusion and occlusion reduction techniques [12]. However, this section focuses on approaches that can specifically be applied to transportation networks. Figure 1 shows an overview and visual comparison of existing OMTs. Besides reducing the height of occluders (Figure 1(c)), the remainder of this section discusses further approaches occlusion reduction (Figure 1(a)).

2.1 Transparency Rendering Techniques

The traditional approach for rendering transparent objects over an opaque scene (Figure 1(c)) requires the scene objects ordered first [15] and than back-to-front rendering to frame buffer with α-blending enabled [2]. Due to the resulting impact on rendering performance and limitations w.r.t. scene representations it is not

VINCI'19, September 2019, Shanghai, China

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn



(a) Occlusion. (b) Flattening. (c) Transparency. (d) Ghosted View. (e) Cut-away View. (f) Fence Hint. (g) Outline Hint.

Figure 1: Comparison of occlusion management techniques applied to a street that is occluded by a 3D building object.

used. Today, one can distinguish basically between two GPU-based approaches to transparency rendering: *order-independent* and *stochastic*, both varying in run-time and space performance, as well as output quality. Specific implementations for Order-independet Transparency (OIT) techniques changed significantly over recent years.

As a first approach, *depth-peeling* [13] requires Render-to-Texture (RTT) functionality in combination with Mutliple Render Targets (MRTs) [2] of the same object to peel-away as well as textures to store layers of unique depth complexity, which can be processed are subsequently combined into a final rendering. The run-time complexity of the approach as well as the space complexity for the layer representation directly correlates with the scene complexity [9], i.e., , it is hardly useable for scenes of high geometric complexity. To reduce the number of passes required, dual depth-peeling [5, 19] uses high precision floating point textures with respective arithmetic to peel two layers of unique depth complexity simultaneously. To increase flexibility during compositing operations, K-Buffer [4] or stencil-routed A-Buffer[23] can be used.

2.2 Ghosted and Cut-Away Views

Occluded scene parts can be visualized by rendering the occluding surfaces semi-transparent [11] (Figure 1(d)). The portion of occluding elements that has to be rendered in such a way is computed according to the position of the objects that should be visible, so that only the necessary parts are rendered with ghosted views. Instead of removing scene elements, they can be presented as so-called *ghosts*, i.e., they are rendered using transparency or wire-framing [7]. The usage of transparency retains structural aspects, but allows for partial depiction of obstructing elements, while wire-frame rendering show only the salient contours of these elements. This basically preserves structural aspects, but allowing the possibly obstructed elements to be almost completely visible.

A detailed work on cut-away views (Figure 1(e)) is presented in [17]. It shows how removing outside-parts of a 3D scene, or use transparent rendering to reveal occluded parts, which in several cases is more important. This approach finds an application for example in anatomy: to analyze the interior of the human body many layers need to be removed in order to inspect a 3D model. Another approach to cut-away views is the one presented by Lidal *et al.* : the 3D scene is cut by means of clipping planes [34] or simple clipping solids [18]. View-dependent cut-away views are presented by Burns *et al.* , which are based on clipping surfaces that cut-off the occluding elements according to the shape of the Object-of-Interests (OOIs) to retain the maximum amount of context possible [7]. Cut-away views allow for removal of scene elements to simplify depiction and to hide unnecessary objects that might distract the attention from OOI.

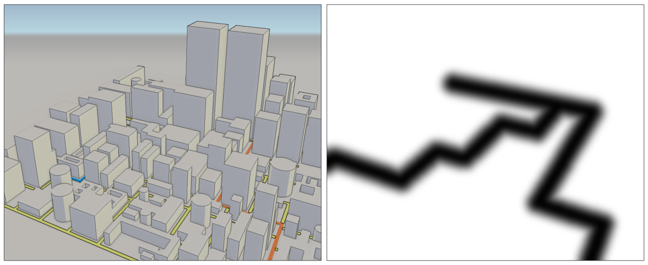
2.3 Distortion and Multi-perspective Views

Besides transparency and cut-away views, occlusion reduction can also be performed by configuring and applying rendering techniques for *panoramic imaging*, *non-linear perspectives*, and *global deformations*, which are discussed for the sake of completeness in the remainder of this section.

Panoramic Imaging. Panoramic maps, introduced by H.C. Berann, combine different projection techniques to generate maps that facilitate user orientation by reducing occlusions, among others. Premoze introduced a framework for the computer aided generation of panoramic maps [26]. Takahashi *et al.* present a semi-automatic approach to generate panoramic maps that relies on global deformations [32]. Falk *et al.* introduced a semi-automatic technique based on a force field that is extracted from the terrain surface [14]. Degener & Klein focus on parameters such as occlusion and feature visibility in their automatic generation of panoramic maps [10]. All approaches combine non-linear perspectives in one final image, but rely on different techniques.

Non-linear Perspectives. Non-linear perspectives can be achieved with different techniques: (1) using non-standard, non-linear projection to produce a non-linear perspective image, or combine several images taken from different perspectives [1, 31]; (2) reflections on non planar surfaces, and (3) local or global space deformation [36]. The combination of different images to one final image as used in [1] and [31] can also be expressed by a space-deformation as introduced in [3]. The Single Camera Flexible Projection Framework of Brosz *et al.* is capable of combining linear, non-linear and hand-made projections in real-time [6]. The projections are described by a deformed viewing volume. Similar to free-form deformations [30], the view frustum serves as lattice. Objects or viewing rays are deformed according to the deformation of the lattice. For the occlusion free visualization of driving routes, Takahashi *et al.* rely on global space deformation [33]. On the one hand the techniques described above offer a broad and flexible definition of the projections, which enables the user to control nearly every facet of the final perspective. On the other hand a large number of non-intuitive parameters have to be controlled.

Global Deformations. The work of Lorenz *et al.* use global deformation to generate non-linear perspectives [20]. The geometry is mapped on two different planes, which are connected by a Bezier surface. The planes may vary in tilt, allowing for a combination



(a) Rendering w. occlusion. (b) Distance-field of routes.

Figure 2: Reference image for the occlusion management techniques discussed in this paper (a) and the distance field generated for the highlighted routes of the transportation network (b), used by the occluder flattening, transparency, as well as ghosted and cut-away views.

of two different perspectives. Similar to panoramic maps, a mixture of cartographic maps and aerial images is used. The different stylization are seamlessly blended in the transition between the planes. Möser *et al.* extend this idea by using a more flexible Hermite curve to control the deformation [22]. They also rely on a combination of aerial and cartographic images to apply a kind of generalization in the more distant parts of the scene. Further, Rademacher introduces a view-dependent geometry [27] by defining key-deformation with associated key viewing points. A similar approach is used by Burtnyk *et al.* to implement interactive stylized camera control [8]. Martín *et al.* present another view-dependent variation of global deformations [21] that can be modified by a view or distant-dependent control function.

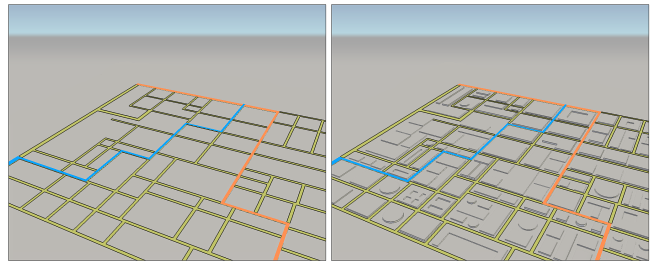
3 OCCLUSION MANAGEMENT TECHNIQUES

This section presents an overview of OMTs that can be applied to transportation networks within virtual 3D city models.

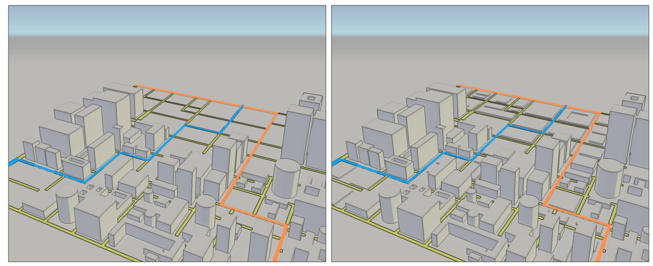
3.1 Preliminaries and Assumptions

Our approach makes assumptions of the representation of the transportation network data for rendering. We assume that a network comprises a number of edges (line segments) referring to its position and additional node attributes. Further, the network representation should also encode adjacency information [35]. Figure 2 shows a reference image for the occlusion management techniques discussed in this paper (Figure 2(a)) and the distance field generated for the highlighted route of the transportation network (Figure 2(b)) (obtained using *Jump-Flooding* [28]), used by the occluder flattening, transparency, as well as ghosted and cut-away views.

For the particular rendering techniques implementing the respective occlusion management approaches, we make the following assumptions on the scene structure: (1) the 3D digital terrain model and the geometric representation of the transportation network can be rendered separately from building objects, (2) network parts of particular interest can be rendered separately, as well as (3) for per-object adaptive approaches access to the individual bounding volumes (e.g., an Axis-Aligned Bounding Box (AABB)) associated with an object ID is required (represented as a vertex attribute).



(a) Global without footprint. (b) Global with footprint.



(c) Adaptive without footprint. (d) Adaptive with footprint.

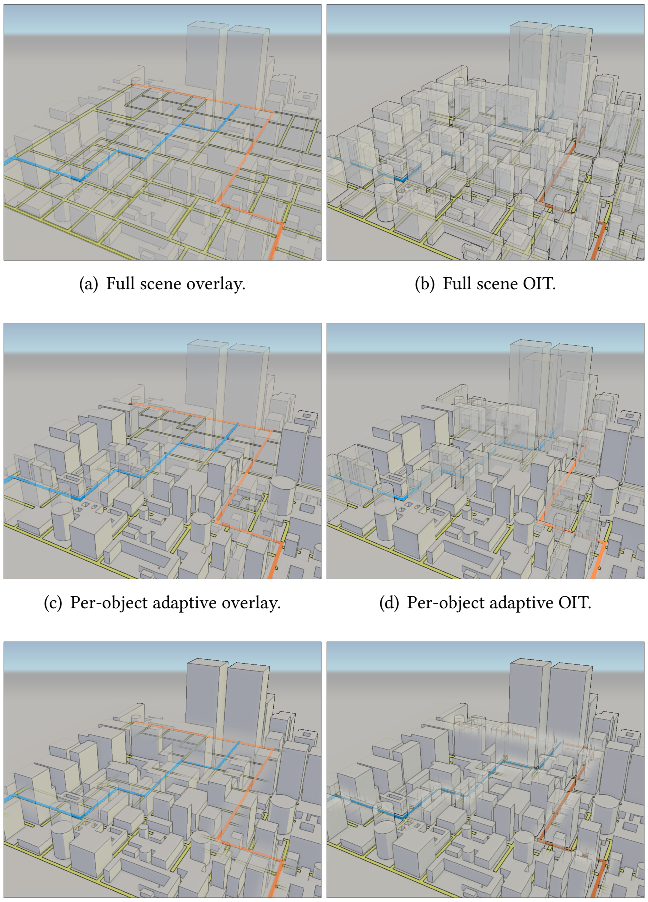
Figure 3: Comparison of global (top) and adaptive (bottom) occluder flattening without (left) and with (right) rendering of building footprints.

Further, the 3D city model is assumed to be planar, since not all of the discussed OMTs are applicable to mountainous regions.

3.2 Occluder Flattening

Synopsis. The basic idea of the Occluder Flattening (OF) approach is to basically avoid occlusions by simply flattening the occluded objects from the scene. Figure 3 shows an overview of using *global* and *adaptive flatting* of scene occluders (e.g., building objects) for occlusion management. However, completely omitting rendering of occluders on a global level (Figure 3(a)) also removes any hints for the viewer about the occluders shape, e.g., its ground plan. This is approached by flatting occluder geometry to a height value close to the virtual terrain model (Figure 3(b)). Instead of performing flattening globally, an adaptive approach allows for conveying more context information (Figs. 3(c) to 3(d)).

Rendering Technique. For the *global* occluder flatting mode, all building objects are omitted of rendering (Figure 3(a)) or scaled to a fixed height using a vertex shader. Alternatively, *adaptive* occluder flatting can be implemented as follows. First, occlusion detection is performed on a per-building level. Therefore, for each building AABB, intersection with all route segments is tested using the projected and transformed coordinates obtained by virtual camera configuration. The result if an object occludes a route segment is stored in an occluder map comprising the object ID and a Boolean value. Following to that, the digital terrain model including the complete transportation network is rendered to the framebuffer. Finally, the building objects are rendered using the occluder map. Using the respective object ID, a vertex shader is used to vertically scale all occluder buildings to a fixed height prior to rasterization



(a) Full scene overlay. (b) Full scene OIT.

(c) Per-object adaptive overlay. (d) Per-object adaptive OIT.

(e) Per-pixel adaptive overlay. (f) Per-pixel adaptive OIT.

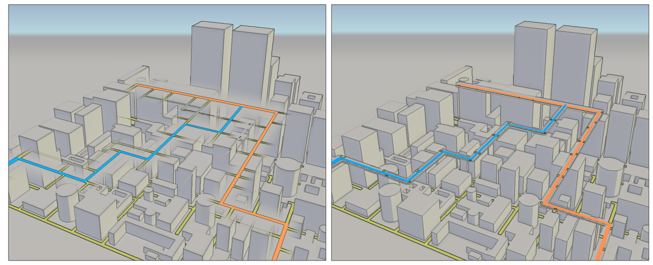
Figure 4: Comparison of global (top row), per-object-adaptive (middle row), and per-pixel-adaptive (bottom row) approach using transparency overlay (left) or OIT (right).

(Figure 3(d)) Alternatively, all respective fragments are discarded using a fragment program (Figure 3(c)).

Discussion. Simply omitting the rendering of occluders is the simplest and straight forward approach to occlusion management. However, context information is almost completely lost. In an abstract visualization (e.g., without facade textures) a user can hardly distinguish if the OMT modifies the building height or if it is the building's original height is small, thus ambiguous. Further, during changes of the virtual camera, the visualization is not temporal coherent, due to buildings popping up or suddenly disappearing (popping artifacts), which also impact depth perception.

3.3 Occluder Transparency Rendering

Synopsis. Figure 4 shows an overview of using *global*, *per-object adaptive*, and *per-pixel adaptive* transparency rendering of scene occluders (e.g., building object) for occlusion management. As shown, there are two approaches for using transparency in occlusion management: *overlay mode* and *transparency mode*. Figure 4(a) shows a simple blending between the rendering of virtual 3D city model and onto of the transportation network, Figure 4(b) shows the blending results using OIT. This can be extended by using an *object* (Figs. 4(c) to 4(d)) and *pixel-adaptive* (Figs. 4(e) to 4(f)) approach, respectively.



(a) Ghosted View. (b) Cut-away View.

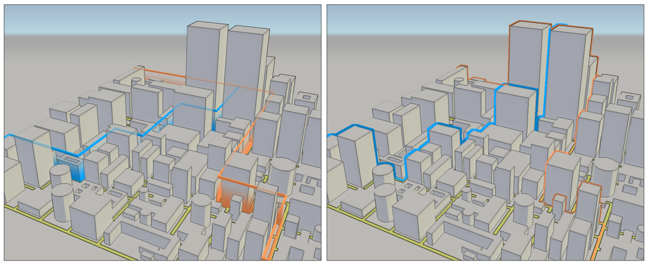
Figure 5: Comparison of using ghosted and cut-away occlusion management techniques.

Rendering Techniques. Rendering transparency-based OMT comprises three major steps: first, rendering the terrain model as well as the complete transportation network and its highlighted parts into the framebuffer. This step is common for both, the overlay and the transparency mode. Following to that, the 3D virtual city model is rendered using RTT in separate G-Buffer (overlay mode) or K-Buffer (transparency mode) respectively. For the object-adaptive mode, this step stores occluder information for each building (similar to the flattening approach) in addition to the color, normal, and depth information. Following thereto, to obtained buffer are blended over the framebuffer contents. For the overlay mode, the blending weight is constant (full scene overlay), derived from the occluder information (per-object adaptive), or from the distance-field of highlighted paths (per-pixel adative). Similarly, the OIT compositing for the transparency mode is performed.

Discussion. While using transparency potentially yields a coherent visualization of both the network and the occluder, it is not suited for virtual 3D city models of dense building placement, especially if viewed using bird's eye perspectives. Transparency conveys individual shapes and the inter-object relations of occluders. In comparison to the remaining techniques in this section, the overlay rendering technique is simple and straight-forward to implement and fast to compute. However, the highlighted paths often appear to "swim" over the terrain model. A major downside of full scene transparency as well as as the per-object and per-pixel adaptive transparency techniques is a potentially high overdraw, which results in invisible paths for scene with high depth complexity. While full scene overlay and transparency techniques behaves temporally consistent, the per-object adaptive approaches are subjected to popping artifacts. Using fading animation counterbalancing this, however, can lead to additional distraction of the viewer.

3.4 Ghosted and Cut-Away Views

Synopsis. The previously described approach of per-pixel adaptive transparency can be considered as instance of ghosted views.



(a) Fence Occlusion Hint. (b) Outline Occlusion Hint.

Figure 6: Comparison of occlusion hint techniques for two highlighted paths of a transportation network.

Figure 5 shows a comparison between an *ghosted view* (Figure 5(a)) and a *cut-away view* (Figure 5(b)).

Rendering Technique. As reviewed in Section 2.2, there are various approaches for the rendering of interactive cut-away views; differing w.r.t. visual quality, rendering performance, and implementation complexity in particular. Since the geometry for the path-of-interest can be characterized as simple (e.g., lines, ribbon alike, or box-shaped) as well as the respective cuts, we propose a straight forward, yet efficient approach for rendering cut-away views for transportation networks that basically comprises four rendering passes.

First, the digital terrain model with the complete transportation network is rendered to the framebuffer. Subsequently, the paths-of-interest are rendered offscreen into a combined color and depth texture buffer. Following to that, only depth component of this texture is dilated: depth values are propagated within a certain radius, which determines the cut width. Finally, the remaining object of the virtual 3D city model are rendered to the framebuffer as follows: fragments whose depth value is less or equal than the dilated depth values (stored in the texture and accessed by screen-space coordinates of the respective fragment) are discarded.

Discussion. In contrast to transparency techniques and similar to flattening techniques, ghosted as well as cut-away views enable a mostly uncluttered views of the OOI.

3.5 Fence Occlusion Hints

Synopsis. Figure 6(a) shows an example of a FOH. Its concept basically comprises the computation of a vertical fence-like representation on top of the original street segment. The main parameters are the fence height h that can vary among the line segments and a global opacity parameter α that controls how much of the fence occludes the remaining scene.

Rendering Technique. The rendering of FOHs basically requires two passes. First, the virtual 3D city model including the transportation network is rasterized into the frame buffer. Subsequently, only the parts of the transportation network to highlight is rasterized. During this step, the vertical fences are computed for each highlighted line segment $L = (V_0, V_1)$ using geometry shader functionality as follows. Given the normalized up-vector N, the four corner points of the fence segment F are determined by $F = (V_0, V_1, V_1', V_0')$, with $V_0' = V_0 \cdot hN$ and $V_1' = V_1 \cdot hN$. For correct blending of the segments to the frame buffer, an OIT technique is used (cf. Section 2.1).

Discussion. FOHs should present sufficient hints to track occluded parts of the highlighted routes. In general, the approach is feasible for urban scenarios, with low variance in building height. However, in virtual 3D city models with a high variance in building height, the approach can decrease coherent depth perception and yield visual clutter. The FOH approach can be discussed according to the even criteria presented in [12] as follows (cf. Table 1):

Primary Purpose (PP): The primary goal of the technique is to reproduce the route and relate it to the surrounding geometry.

Disambiguation Strength (DS): Depending on the height of the fences, visualization can handle inclusion and containment as well as geometry intersection.

Depth Cues (DC): In general, the relation between scene objects is preserved. In rare cases, FOH can impact depth perception.

View Paradigm (VP): The techniques has no restriction with respect to the use of multiple virtual cameras.

Interaction Model (IM): The visualization requires no preprocessing and is designed for interactive use.

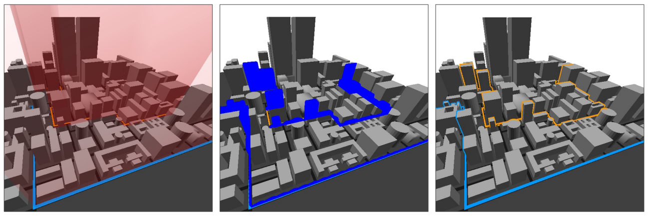
Target Invariances (A/D/G/L): Since the fences cover parts of the scene geometry, the appearance of the scene is not invariant and can impact depth perception. However, geometry and location information of the scene remain invariant.

Solution Space (SS): The FOH implementation works in object space.

3.6 Outline Occlusion Hints

Synopsis. The previously discussed FOH sufficiently communicate the principle direction of the line segments and facilitate the visual separation from background. However, they cannot guaranteed to be not occluded and tend with increasing height occlude more of the scene context as necessary. Here, the novel concept of OOH can be used to counterbalance this shortcoming (Figure 6(b)). Outline hints create a correspondence with the street segments by drawing an outline around the occluder. This technique directly based on concepts described in the previous section. Subsequent to generating fences with a height h that is larger than the highest object in the scene, all fragments that are not within the outline are discarded.

Rendering Technique. The rendering technique is basically an extension on FOH rendering technique. Figure 7 shows the main stages of rendering OOHs. First, the transportation network geometry that should be highlighted is vertically extruded to the maximum height of the scene AABB using geometry shader functionality (Figure 7(a). Subsequently, the resulting geometry is rasterized into a K-Buffer to convey overlapping or self-occluded fence segments. After that, the virtual 3D city model is rendered capturing color and depth information using RTT. Using this depth data, the K-Buffer is processed at each output pixel location, to detect and mark potential fence-scene occlusions within a certain pixel-neighborhood, i.e., thickness (Figure 7(b)) Here, the main parameter is the thickness of the outline t, which can vary with respect to the distance between



(a) Geometry. (b) Fragments. (c) Result.

Figure 7: Stages of rendering outline occlusion hints: the transportation network geometry in focus is vertically extruded and rasterized (a), occluded fragments are identified (b), and filtered to yield the final outline hints (c).

the segment and the virtual camera. Given the minimum and maximum thickness t_{min} and t_{max}, as well as the linear depth $d \in [0, 1]$ of the current fragment F, the resulting outline width can be computed by linear interpolation as follows: $t_F = d \cdot t_{max} + (1 - d) \cdot t_{min}$. Thus, the impression of size-gradient is supported. Following to that, the scene representation and the K-Buffer are merged in a final display pass to yield the outline hints (Figure 7(c)).

Discussion. OOHs supports the user to track highlighted routes of a transportation network. Alternating the brightness of the base color visualizes route turnings and stacked outlines enable the depiction of multiple occluded routes. However, the association between hint and original path by color basically omits the mapping of additional information such as traffic status etc.According to the criteria in [12], this approach can be categorized as follows (cf. Table 1 for comparison).

Primary Purpose (PP): The highlighted route is visualized comprehensibly and its context is preserved, i.e., it is locatable, traceable, and referenced within the scene.

Disambiguation Strength (DS): Enclosed or containing geometry is handled similar to occluded geometry. The technique is independent of the object interaction level.

Depth Cues (DC): The depth relation between scene objects is preserved. The viewer can establish order on different routes w.r.t. depth.

View Paradigm (VP): The technique designed for a single viewing window but makes no restriction to support multiple virtual cameras.

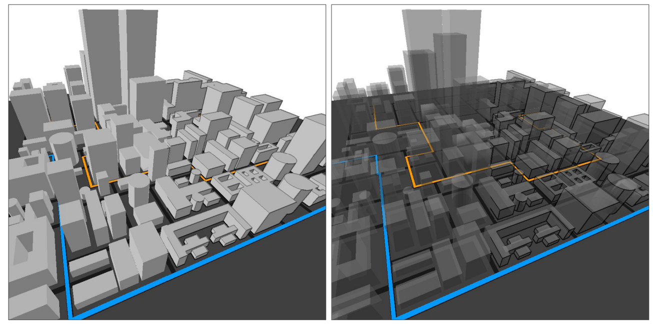
Interaction Model (IM): The technique does not rely on scene pre-processing and can be rendered in real-time for interactive use.

Target Invariances (A/D/G/L): Although the orientation line partially overdraws scene objects, the scene appearance is changed only moderately, i.e., depth-, geometry- , and location information remain invariant.

Solution Space (SS): The OOH implementation works in image space.

4 INTERACTIVE RENDERING TECHNIQUES

This section presents a prototypical implementation of the concepts described previously. The hardware-accelerated rendering techniques are based on OpenGL and enables image synthesis in real-time. It basically comprises three stages that are described in the remainder of this section.



(a) Opaque flat-shading only. (b) Shading and transparency.

Figure 8: Comparison of edge-enhancement impact (right image half) on scene perception for flat-shaded (a) and transparency renderings (b).

4.1 Scene Rendering and Stylization

This stage basically generates a depth-map required for occlusion detection and the color-map comprising the visible geometry of the virtual 3D city as well as features of the transportation network that are not highlighted. The generation is performed within a single rendering pass using render-to-texture with multiple render-targets and thus is efficient for complex geometric scene representations [29]. To enable later efficient depth-testing, linear depth values [16] are stored in a 32 bit floating-point texture that is accessible in the following stages. The color texture is used in a subsequent compositing pass only. This image-based approach enables the application of our concept to different types of 3D geovirtual environments and facilitates the integration into existing rendering and visualization frameworks.

In this forward rendering pass, also shading is applied. For reasons of simplicity and rendering performance, we use cube map texturing [2] with a cube map texture of 1×1 pixels and nearest filtering set. For the renderings depicted in Section 3, the colors for each cube-map face are acquired from Google Maps, but can be arbitrary, e.g., different, distinguishable gray-scale values. We also considered to support image-based edge-enhancement techniques to improve the distinctness of building objects when using flat shading. Figure 8 shows the impact on perception by comparing a flat-shaded scene and its transparent variant with respect to edge enhancement. We choose to incorporate throughout all occlusion management techniques. Especially for transparency rendering, edge enhancement facilitates the perception of object boundaries [24].

4.2 Combination of Management Techniques

Table 1 characterizes the proposed OMTs regarding the seven criteria presented in [12]. Based on the implementation of different OMTs and its respective limitations, we evaluate the potential of combining these accordingly to counterbalance each other.

Table 1: Characterization of the presented occlusion-hint techniques according to the criteria presented in [12].

	PP	DS	DC	VP	IM	A/D/G/L	SS
FOH	relation	contain	very high	single	pass/on	N/Y/Y/Y	object
OOH	relation	intersect	high	single	pass/on	N/N/Y/Y	image

Overview of Combination Approach. While there are different possibilities to approach such combination, we strive for an image-based technique for the sake of simplicity in implementation and flexibility in combination of different types. Therefore, the results of OMTs are stored as images that are blended according to the values of a *combination mask* and displayed as follows:

OMT-Rendering Steps: This renders the OMTs results that should be combined into separate G-Buffer [29] (cf. Section 4.1). It is sufficient to respect the resulting color only, or depending on the mask generation step, depth and stencil values as well. This step is not limited to two OMT.

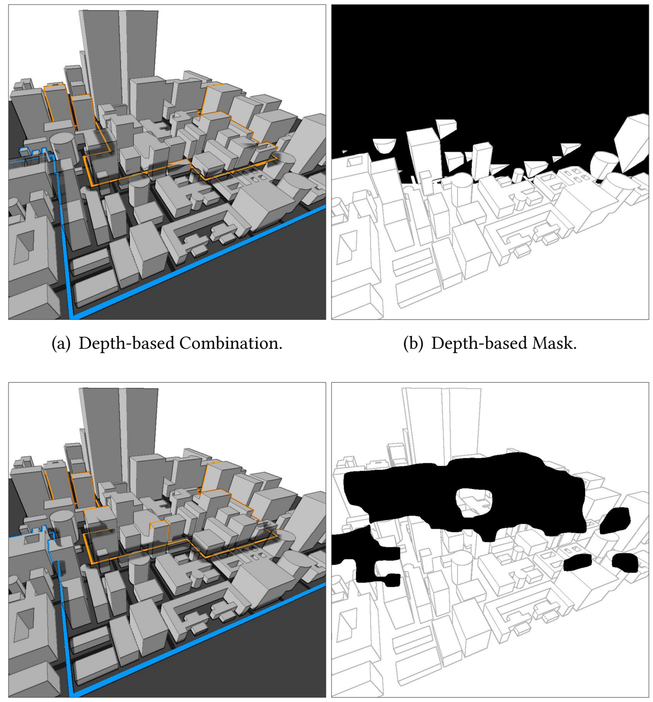
Mask-Generation Step: This steps generates a combination mask that controls how the results of the previous steps are composted into a single output result on a per-fragment basis. For the sake of simplicity, we use a *binary* mask for combining the results of two OMT. The remainder of this section briefly discusses two approaches for mask generation: *depth-based* and *overdraw-based*.

Blending and Display Step: Finally, this step combines the color buffers of each OMT acquired in the first step based on the combination mask and displays the result.

Depth-based Combination Approach. Figure 9(a) shows an example of combining ghosted views and the outline technique based on depth-values. To compute the depth-based mask, first linearized depth values [16] are rendered to a 32 bit depth-buffer texture representation using forward rendering of the complete 3D scene. Subsequently, given a threshold value for the depth, the depth texture values are converted to a binary values prior to blending.

Overdraw-based Combination Approach. Figure 9(c) shows an example example of combining ghosted views and the outline technique based on overdraw represented by stencil-buffer values. In Figure 9(d), the visualization of overdraw show dark areas show high amount of overdraw. The value of overdraw is obtain using rendering with stencil buffer enabled and configured accordingly, and subsequently converted into to binary mask. However, Figure 10 shows the widening of an overdraw input mask via morphological closing (dilation followed by erosion) applied as post-processing step.

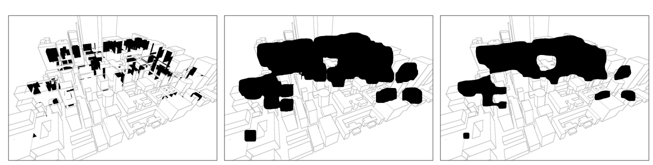
Discussion on Combination Approaches. The implemented combination approaches show that the respective disadvantages of a visualization can be reduced by the use of other techniques. At the same time, the visual complexity of the depicted scene increases. Further, each technique has its own set of additional graphic elements and visual language. If the combination approach is used carelessly, the respective effect is diminished and the clarity is rather reduced, since a stimulus overload deprives the viewer of his focus. Therefore, care must be taken to ensure that each additional visualization provides a sufficient benefit. If the combined visualizations are also extremely different in their effect, improper transitions disrupt the objective of viewing the path. It is also interesting to note that this problem can also occur with a small number of techniques. If, for example, three OMTs are used in too confined a space, or if the transitions from one mode of representation to the next are too close to each other and their change too rapidly, this also results in a overload of stimuli and confusion.



(a) Depth-based Combination. (b) Depth-based Mask.

(c) Overdraw-based Combination. (d) Overdraw-based Mask.

Figure 9: Exemplary combination of ghosted-view and outline hints based on scene depth (top) and scene overdraw (bottom) values.



(a) Input Mask. (b) Dilated Mask. (c) Eroded Mask.

Figure 10: Processing step for computing an overdraw mask based on morphological closing (all mask images are overlayed with scene edges for clarity).

5 CONCLUSIONS AND FUTURE RESEARCH

This paper presents the novel concept of occlusions hints for the visualization of transportation networks in virtual 3D city models. In contrast to existing occlusion-reduction techniques that depict the full visual representation of the occluded object, occlusion hints only visualize cues that indicate if an object is occluded. These cues are attached to occluder geometry and enable a user to infer the route of occluded geometry.

There are several ways to proceed for future research. Qualitative and quantitative user studies should be performed in order to evaluate and refine the concept and its implementation. Prior to that, prototypical implementation can be transferred to mobile devices or adapted for an integration into service-based rendering frameworks. Further, the combination of occlusion hints within multi-perspective views can be researched as well as the integration of suitable labeling techniques for occlusion hints.

ACKNOWLEDGMENTS

We like to thank the anonymous reviewers for their value comments. This work was funded by the German Federal Ministry of Education and Research (BMBF) in the InnoProfile Transfer group "4DnDVis".

REFERENCES

[1] Maneesh Agrawala, Denis Zorin, and Tamara Munzner. 2000. Artistic Multiprojection Rendering. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*. Springer-Verlag, London, UK, 125–136.

[2] Tomas Akenine-Möller, Eric Haines, Naty Hoffman, Angelo Pesce, Michał Iwanicki, and Sébastien Hillaire. 2018. *Real-Time Rendering 4th Edition*. A K Peters/CRC Press, Boca Raton, FL, USA. 1200 pages.

[3] Alan H. Barr. 1984. Global and local deformations of solid primitives. In *SIGGRAPH '84*. ACM, New York, NY, USA, 21–30. https://doi.org/10.1145/800031.808573

[4] Louis Bavoil, Steven P. Callahan, Aaron Lefohn, João L. D. Comba, and Cláudio T. Silva. 2007. Multi-fragment Effects on the GPU Using the K-buffer. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games (I3D '07)*. ACM, New York, NY, USA, 97–104. https://doi.org/10.1145/1230100.1230117

[5] Louis Bavoil and Kevin Myers. 2008. *Order Independent Transparency with Dual Depth Peeling*. Technical Report. NVIDIA OpenGL Applications Engineering.

[6] John Brosz, Faramarz F. Samavati, M. Sheelagh, T. Carpendale, and Mario Costa Sousa. 2007. Single Camera Flexible Projection. In *NPAR '07: Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*. ACM Press, New York, NY, USA, 33–42. https://doi.org/10.1145/1274871.1274876

[7] Michael Burns and Adam Finkelstein. 2008. Adaptive Cutaways for Comprehensible Rendering of Polygonal Scenes. In *ACM SIGGRAPH Asia 2008 Papers (SIGGRAPH Asia '08)*. ACM, New York, NY, USA, Article 154, 7 pages.

[8] Nicholas Burtnyk, Azam Khan, George Fitzmaurice, Ravin Balakrishnan, and Gordon Kurtenbach. 2002. StyleCam: interactive stylized 3D navigation using integrated spatial & temporal controls. In *UIST '02*. ACM, New York, NY, USA, 101–110. https://doi.org/10.1145/571985.572000

[9] Nathan Carr, Radomir Mech, and Gavin Miller. 2008. Coherent Layer Peeling for Transparent High-Depth-Complexity Scenes. In *Graphics Hardware*, David Luebke and John Owens (Eds.). The Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 33–40. https://doi.org/10.2312/EGGH/EGGH08/033-040

[10] Patrick Degener and Reinhard Klein. 2009. A Variational Approach for Automatic Generation of Panoramic Maps. *ACM Trans. Graph.* 28, 1 (2009), 1–14. https://doi.org/10.1145/1477926.1477928

[11] Niklas Elmqvist, Ulf Assarsson, and Philippas Tsigas. 2007. Employing Dynamic Transparency for 3D Occlusion Management: Design Issues and Evaluation. In *Human-Computer Interaction–INTERACT 2007*. Springer, 532–545.

[12] Niklas Elmqvist and Philippas Tsigas. 2008. A Taxonomy of 3D Occlusion Management for Visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 5 (2008), 1095–1109. https://doi.org/10.1109/TVCG.2008.59

[13] Cass Everitt. 2001. *Interactive Order-Independent Transparency*. Technical Report. NVIDIA OpenGL Applications Engineering.

[14] Martin Falk, Tobias Schafhitzel, Daniel Weiskopf, and Thomas Ertl. 2007. Panorama maps with non-linear ray tracing. In *GRAPHITE '07*. ACM, New York, NY, USA, 9–16. https://doi.org/10.1145/1321261.1321263

[15] Naga K. Govindaraju, Michael Henson, Ming C. Lin, and Dinesh Manocha. 2005. Interactive visibility ordering and transparency computations among geometric primitives in complex environments. *Proceedings of the 2005 symposium on Interactive 3D graphics and games - SI3D '05* 1, 212 (2005), 49. https://doi.org/10.1145/1053427.1053435

[16] Eugene Lapidous, Guofang Jiao, Jianbo Zhang, and Timothy Wilson. 2001. Quasi-linear Depth Buffers with Variable Resolution. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware (HWWS '01)*. ACM, New York, NY, USA, 81–86. https://doi.org/10.1145/383507.383530

[17] Wilmot Li, Lincoln Ritter, Maneesh Agrawala, Brian Curless, and David Salesin. 2007. Interactive Cutaway Illustrations of Complex 3D Models. *ACM Trans. Graph.* 26, 3, Article 31 (July 2007). https://doi.org/10.1145/1276377.1276416

[18] Endre M. Lidal, Helwig Hauser, and Ivan Viola. 2012. Design Principles for Cutaway Visualization of Geological Models. In *Proceedings of the 28th Spring Conference on Computer Graphics (SCCG '12)*. ACM, New York, NY, USA, 47–54. https://doi.org/10.1145/2448531.2448537

[19] Fang Liu, Meng-Cheng Huang, Xue-Hui Liu, and En-Hua Wu. 2009. Efficient Depth Peeling via Bucket Sort. In *Proceedings of the Conference on High Performance Graphics 2009 (HPG '09)*. ACM, New York, NY, USA, 51–57. https://doi.org/10.1145/1572769.1572779

[20] Haik Lorenz and Jürgen Döllner. 2008. Dynamic Mesh Refinement on GPU using Geometry Shaders. In *Proceedings of the 16th International Conference on Computer Graphics, Visualization and Computer Vision (WSCG2008)*. UNION Agency - Science Press, Plzen, Czech Republic, 97–104. Online Proceedings.

[21] D. Martín, S. García, and J. C. Torres. 2000. Observer dependent deformations in illustration. In *NPAR '00*. ACM, New York, NY, USA, 75–82. https://doi.org/10.1145/340916.340926

[22] Sebastian Möser, Patrick Degener, Roland Wahl, and Reinhard Klein. 2008. Context Aware Terrain Visualization for Wayfinding and Navigation. *Comput. Graph. Forum* 27 (2008), 1853–1860. https://doi.org/10.1111/j.1467-8659.2008.01332.x

[23] Kevin Myers and Louis Bavoil. 2007. Stencil Routed A-Buffer. In *ACM SIGGRAPH 2007 Sketches (SIGGRAPH '07)*. ACM, New York, NY, USA, Article 21, 1 pages. https://doi.org/10.1145/1278780.1278806

[24] Marc Nienhaus and Jürgen Döllner. 2005. Blueprint Rendering and Sketchy Drawings. In *GPU Gems II: Programming Techniques for High Performance Graphics and General-Purpose Computation*, M. Pharr (Ed.). Addison-Wesley Professional, 235–252.

[25] Sebastian Pasewaldt, Amir Semmo, Matthias Trapp, and Jürgen Döllner. 2012. Towards Comprehensible Digital 3D Maps. In *Service-Oriented Mapping 2012 (SOMAP2012)*, Markus Jobst (Ed.). Jobstmedia Management Verlag, Wien, 261–276.

[26] Simon Premoze. 2002. Computer Generated Panorama Maps. In *ICA Mountain Cartography Workshop*. Online, Mt. Hood, Oregon, USA, 4.

[27] Paul Rademacher. 1999. View-dependent geometry. In *SIGGRAPH '99*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 439–446. https://doi.org/10.1145/311535.311612

[28] Guodong Rong and Tiow-Seng Tan. 2006. Jump Flooding in GPU with Applications to Voronoi Diagram and Distance Transform. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games (I3D '06)*. ACM, New York, NY, USA, 109–116. https://doi.org/10.1145/1111411.1111431

[29] Takafumi Saito and Tokiichiro Takahashi. 1990. Comprehensible Rendering of 3-D Shapes. *SIGGRAPH Comput. Graph.* 24, 4 (Sept. 1990), 197–206. https://doi.org/10.1145/97880.97901

[30] Thomas W. Sederberg and Scott R. Parry. 1986. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.* 20, 4 (1986), 151–160. https://doi.org/10.1145/15886.15903

[31] Karan Singh. 2002. A Fresh Perspective. In *Proceedings of the Graphics Interface 2002 Conference, May 27-29, 2002, Calgary, Alberta, Canada*. Canadian Human-Computer Communications Society (CHCCS), Calgary, Alberta, Canada, 17–24. https://doi.org/10.20380/GI2002.03

[32] Shigeo Takahashi, Naoya Ohta, Hiroko Nakamura, Yuriko Takeshima, and Issei Fujishiro. 2002. Modeling Surperspective Projection of Landscapes for Geographical Guide-Map Generation. *Computer Graphics Forum* 21 (2002). Issue 3. https://doi.org/10.1111/1467-8659.t01-1-00585

[33] Shigeo Takahashi, Kenichi Yoshida, Kenji Shimada, and Tomoyuki Nishita. 2006. Occlusion-Free Animation of Driving Routes for Car Navigation Systems. *IEEE TVCG* 12 (2006), 1141–1148. https://doi.org/10.1109/TVCG.2006.167

[34] Matthias Trapp and Jürgen Döllner. 2013. 2.5D Clip-Surfaces for Technical Visualization. *Journal of WSCG* 21 (2013), 89–96.

[35] Matthias Trapp, Amir Semmo, and Jürgen Döllner. 2015. Interactive Rendering and Stylization of Transportation Networks Using Distance Fields. In *Proceedings of the 10th International Conference on Computer Graphics Theory and Applications (GRAPP 2015)*. SCITEPRESS - Science and Technology Publications, Lda, Portugal, 207–219. https://doi.org/10.5220/0005310502070219

[36] Scott Vallance and Paul Calder. 2001. Multi-perspective images for visualisation. In *VIP '01*. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 69–76.